Portable and Scalable In-vehicle Laboratory Instrumentation for the Design of i-ADAS

Jihyun Cho

Portable and Scalable In-vehicle Laboratory Instrumentation for the Design of i-ADAS

(Spine Title: Vehicle Instrumentation for i-ADAS)

(Thesis Format: Monograph)

by

Jihyun Cho

Graduate Program in Computer Science

2

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science

School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario

THE UNIVERSITY OF WESTERN ONTARIO
SCHOOL OF GRADUATE AND POSTDOCTORAL STUDIES

CERTIFICATE OF EXAMINATION

Supervisor	Examiners
Steven S. Beauchemin	Mike A. Bauer
Supervisory Committee	John Barron
	Jim Dickey

The thesis by

Jihyun Cho

entitled:

Portable and Scalable In-vehicle Laboratory Instrumentation for the Design of i-ADAS

is accepted in partial fullfillment of the
requirements for the degree of
Master of Science

Date

Chair of the Thesis Examination Board

Acknowledgement

I would like to first thank my supervisor Steven Beauchemin for giving me an opportunity to work with him on RoadLab. My understanding of the subject was significantly enhanced through his experience, guidance, and support. From the beginning of the project to the end, his encouragement kept me focused on achieving my goals.

I would also like to thank all the members of RoadLab. Special thanks to Taha Kowsari for sharing with me his experience to overcome programming obstacles. I would like to thank Owen McCarthy for his work on CANbus interface.

Abstract

According to the WHO (World Health Organization), world-wide deaths from injuries are projected to rise from 5.1 million in 1990 to 8.4 million in 2020, with traffic-related incidents as the major cause for this increase. Intelligent, Advanced Driving Assistance Systems (i-ADAS) provide a number of solutions to these safety challenges. We developed a scalable in-vehicle mobile i-ADAS research platform for the purpose of traffic context analysis and behavioral prediction designed for understanding fundamental issues in intelligent vehicles. We outline our approach and describe the in-vehicle instrumentation.

Keywords: Advanced Driving Assistance Systems, i-ADAS, In-vehicle research platform

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Advanced Driver Assistance System

Advanced Driver Assistance Systems (ADAS) are a collection of applications that will assist a driver to safely avoid accidents and minimize the consequences of traffic-related incidents. Driving directly affects human lives and health: in 2006, in North America, commuting led to 6 million accidents, 1.7 million injuries, and 39,000 fatalities [1]. Yet the simplest of driving assistance systems, such as enhanced stability control (ESC), may reduce single-vehicle crashes by 29 to 35 percent [7]. Even with low penetration levels of ADAS technologies (five to 10 percent), the safety of every vehicle increases [4].

Hypothetically, a crash-less car could be made much lighter without endangering its occupants. Vehicle curb weight is a significant fuel consumption factor. As of today, such lighter-weight vehicles cannot be manufactured due to enforced crash-safety ratings.

Compounding the problem, traffic congestion is a growing concern worldwide as car ownership continues to increase [4]. In a typical congestion situation, an aerial view of the traffic reveals that cars occupy roughly only 10 percent of the available pavement.

ADAS technologies could improve this radically by automating longitudinal vehicle control, for instance. Such systems could increase the density of traffic, with vehicles following each other closely and safely, reducing the need to extend current highway infrastructures [14].

Also of significance is the fact that the average driving age in western countries is increasing. While this should not be a problem in itself, it has nonetheless been established that the decline in cognitive and motor abilities affects the safety of drivers and others around them [5] [3].

1.2 Intelligent ADAS

We address the physical design and implementation of an in-vehicle laboratory for the development of intelligent, advanced driving assistant systems (i-ADAS). While research on ADAS integrates a number of different functions such as forward collision detection and lane departure tracking [30], little attention was devoted to the monitoring of events and factors that directly concern the driver of the vehicle. It is only recently that cognitive aspects have been considered as a legitimate part of i-ADAS [27]. Since 95 percent of all accidents are caused by human error, it is crucial that these aspects of driving be a central part of i-ADAS [8]. Keeping the driver as an active participant in the feedback mechanisms provides contextually motivated informational support and offers immediate applications for enhancing safety [19].

The extended possibilities of integrated i-ADAS are very relevant research areas as they do not intend to replace the driver as much as to assist in a safer driving process. As has been pointed out by Petersson et al. [19], what remains to be automated to reach the state where vehicles become completely autonomous, in a practical sense, turns out to be difficult and elusive in everyday driving situations. However, it is our belief that driver support through i-ADAS can be deployed more readily, with consequent socioeconomic benefits.

1.3 Challenges

ADAS aim to improve driving comfort and the traffic safety. High-precision real-time processing is required for such systems. Typically, ADAS consist of different detection modules (e.g. pedestrian detection, lane detection, sign detection) associated with the warning system to inform the driver [19] [27].

The accuracy of the detection modules in ADAS has a direct impact on the safety of the vehicle and its passenger. The false positive detection of a stop sign, for example, may bring the vehicle to a sudden stop, which increases the danger of crashing with a following car. On the other hand, drivers may be in an accident due to the misdetection of oncoming danger. Accuracy and the reliability are crucial factors in ADAS.

Generally, a variety of data from different interfaces is used to improve the performance of the detection modules. Glaser *et al.* [11] coupled navigation data with visual data for detecting the lane. McCall and Trivedi [24] make use of vehicle state (e.g. speed, accelorator and brake pedal position) collected through the Controller Area Network (CANbus) and other motion-related sensors in conjunction with LIght Detection And Ranging (lidar) and near-infrared cameras for a brake assistance system. As this previous work has suggested, the infrastructure of ADAS must provide a rich set of data about the vehicle's surroundings for reliable ADAS design.

This means that a vast amount of data is continually sent to the processing unit. This system must be capable of handling the collected data and sending these to appropriate modules in a fashion that will minimize the delay.

Communication between different modules must be taken into the consideration for designing the ADAS. In the case of a detection module that estimates the ground plane, the results can be used in lane, sign, and pedestrian detection modules, and provide a constraint to improve result accuracy. Because one module's results may provide a constraint to other modules, which could greatly increase others' accuracy, ADAS should provide an efficient inter-module communication mechanism.

During the data acquisition and inter-module communication, we must validate data and the results from modules in real-time. Depending on the data and its source, the data may not be valid after a certain length of time, or some data may have higher priority over others. The system must provide a mechanism to support the determination of those decisions.

Since the system is operating on a vehicle, the vibration of the vehicle introduces additional noise in image data. The instrumentation should be selected to minimize the effect of this vibration. Prior to processing the visual data, the module must consider the impact of such noise as well.

Ultimately, the system should be able to monitor and understand the driver's intention and predict the future manoeuvre, and warn the driver in a manner that minimizes driver distractions. To achieve this, we need a way to assess the perception of the driver with respect to the environment of the vehicle.

1.4 Scope of the thesis

This thesis presents the description of the instrumentation of a vehicle for the on-going ADAS development in the RoadLab project. The scope of the paper includes an explanation of the types of devices used in the infrastructure of ADAS, and a description of the software created for initiating ADAS development. In terms of hardware, we will discuss the rationale to the choice of each device and its usage. The devices include visual sensors, a Global Positioning System (GPS) unit, CANbus interface and data processing units. The paper will also discuss the set of software interfaces developed for the RoadLab ADAS: RoadLab Recorder, RoadLab Calibration interface, and RoadLab Sequence Reader. We will discuss each interface's functionality and structure, briefly discussing the underlying methods.

1.5 Thesis Outline

The rest of this thesis can be outlined as follows: Chapter 2 gives overview of the recent work on ADAS and how our work is related. Chapter 3 describes our system infrastructure in detail. Chapter 4 describes our approach on camera calibration and data synchronization. Then the use of the calibrated sequence is shown with stereo computation in chapter 5. Chapter 6 provides the description of the applications created as a part of instrumentation. Chapter 7 describes the data and the format in which we collected it. Chapter 8 and 9 discuss the limitations and the evaluations of the system. Lastly, we sum up our work and suggest possible future extensions.

Chapter 2

Related Work

2.1 Related Literature

While injuries per driven kilometre are in decline in developed countries [1], a reverse trend can be observed elsewhere in the world, especially in regions where car ownership is rising quickly.

Advanced Driving Assistance Systems are generally designed to support decision making by providing ergonomic information on the driving environment, such as the presence of surrounding vehicles, potential hazards, and general traffic conditions. A large array of sensing devices and data fusion strategies have been devised and deployed to create effective ADAS. Sensing may be performed with radar [14], [38], lidar [15], or laser range finders [20].

However, a majority of ADAS rely principally on vision systems supported by other sensor modes [23]. McCall and Trivedi [23] conducted extensive reviews on "video-based lane estimation and tracking" (VioLET) systems. They reported that laser radar sensors can perform well in certain situations, but multiple lanes can only be detected with the aid of visual data. Therefore, visual data play a significant role in most of the lane estimation systems, and data from other sensors are used for refining the result in many detection-related tasks.

To refine and optimize the performance of ADAS, GPS data and additional information about the vehicle motion can be useful as shown in Glaser *et al.* [11] In this work, they developed a lane departure warning system. The instrumentation of their system included motion sensors, an enhanced map, a GPS unit and visual sensors. The additional information from the motion sensors allowed them to take into consideration both the lateral movement and the longitudinal motion.

Another example of a multi-sensored system is [24]. McCall and Trivedi [24] developed a brake assistance system that is aware of driver behaviour and the driving environment. The need for braking is detected based on the laser radar range finder and the data from CANbus (e.g. brake pressure, steering angle, and accelerator position). Driver head and feet positions are also used for detecting such situations, where the driver's head position is monitored with color cameras, and near-infrared cameras are used to monitor the feet. This system suggests the importance of considering the driver as a part of ADAS.

As more attention is devoted to the driver's perception during the operation of the vehicle, a number of researchers suggest different approaches for collecting data. There are two main approaches for data collection: using a simulator [29] [18] and using an instrumented vehicle [26] [2]. One of the advantages of simulators is that it makes the data acquisition process easier in a variety of scenarios. Another advantage of the simulator is that it allows the researcher to focus on a specific scenario. However, due to the dynamic nature of the driving environment, a simulator may not be sufficient to create scenes (e.g. vibration of the vehicle and illumination effects) for testing and developing the complete ADAS.

Perez *et al.* [26] developed an in-vehicle data recorder. Their system is equipped with up to eight medium-resolution cameras and one high-resolution camera. The radar scanning system is used in the front of their test vehicle. They equipped a

24 GHz Doppler sensor to the underbody of the car for accurate speed measure. Ultrasonic sensors are attached on both sides of the vehicle to measure the free space around the vehicle. An eye-tracking system monitors the driver's gaze and observes the viewpoint. The system aims to record real-time data of the vehicle's internal and external state, and analyze driver behaviour in real driving situations.

Another form of widely used data is from event data recorders (EDRs). EDRs are used in real vehicles to collect data. The data from these systems contain certain information (e.g. engine faults, and a sudden change in wheel speed) from before, during, and after a vehicle collision. These data typically do not contain video, sound, location or any other external conditions. The United States National Highway Traffic Safety Administration (NHTSA) uses EDRs for their crash investigations.

Data collection and processing must be performed in real time. In order to meet this hard real-time requirement, early ADAS deployed special parallel architectures using transputers [9], a massively paralleled Single instruction, multiple data (SIMD) machine [6] and other computers. Even with the dramatic improvement of the computing power over the last decade, complete ADAS with single personal computer (PC) or a processor is not practical. A processing unit composed of multiple PCs and custom hardware [34] [36] is a very common approach. In addition, the system can take an advantage of graphics processing unit's (GPU) parallel architecture to improve its performance [31].

2.2 Our contribution

Our approach, while sharing elements with other's work, is unique in several ways. First, we designed a portable instrumentation requiring no modification to the vehicular platform, using low-cost, off-the-shelf components that are widely available. Second, our on-board computational approach rests on scalability. That is to say, additional computing power can easily be added to the current instrumentation, without any modifications to the existing system. This, of course, is a core requirement, as algorithms must run in real- time. Third, the data collected by the in-vehicle laboratory will be available to the research community. The infrastructure of the ADAS research platform we are introducing will encourage other researchers join the field and minimize the time and effort required in investigating ADAS or other related fields.

Chapter 3

In-vehicle Laboratory

3.1 System Overview

The design of the instrumented vehicle follows principles of sensor portability and computing scalability. Sensor portability is achieved by using vacuum devices to attach the instrumentation equipment, such as stereo camera rigs, LCD screens, and GPS units, to the interior glass and external metal surfaces of the vehicle, without performing permanent modifications to the vehicle. The odometry is obtained from the On-Board Diagnostic systems (OBD-II) outlet located under the dashboard on the driver's side of the vehicle. Each minute, the sensory equipment sends two to six GBytes of data depending on the number of visual sensors equipped, to the on-board computer. With such large amounts of data to process, the computing equipment was designed with scalability as a guiding principle. For this purpose, a disk-less cluster arrangement was chosen essentially to provide the option of adding computing nodes as necessary. Currently, the on-board computer is composed of 16 computing nodes distributed over four boards networked with a gigabit switch. The nodes and the switch are inside a portable server case, which in turn can be installed on the back seat or in the trunk of the vehicle. The computer and instrumentation are powered with by 1500W inverter connected directly to the vehicle battery. The instrumentation can be run continually without battery drainage.

3.2 Visual Sensors

Most ADAS heavily rely on the laser or lidar sensors to achieve their accuracy. These sensors are costly and the high cost of ADAS is a constraint to the public market. To overcome this constraint, low-cost, off-the-shelf cameras are selected as visual sensors.

The visual sensors on the vehicle should appropriately monitor the immediate environment (lanes, other vehicles, pedestrians, obstacles, etc). These hardware systems must be capable of high sampling rates (30Hz or more) such that sufficient accuracy in image processing and automated vision processes is achieved. It is useful to keep in mind that the position of a vehicle moving at 120 kph changes by 33 metres every second. The latency between the cameras and the data processing unit should be minimal. Firewire cameras minimize the latency of the data transfer. The 1394a firewire bus supports 400Mbits/s while the 1394b firewire bus provides up to 800Mbits/s.

For these reasons, we have chosen "Grasshopper" cameras from Point Grey (see Figure 3.1). The Grasshopper supports both 400Mbits and 800Mbits bus with a high frame rate. The camera can also be synchronized when multiple cameras are connected to the same bus, simplifying synchronization issues for the cameras. For scalability and the future extension, the image frames include time stamps that resolves synchronization issues with other types of sensors or other camera products. All the image frames from visual sensors are synchronized to within 125 μs. Once synchronized frames are obtained, stereo depth maps are computed at frame rate, based on the calibration parameters.

In order to achieve full coverage of the surrounding environment of the vehicle lab, multiple cameras with different ranges are required. For these visual sensors, it is crit-

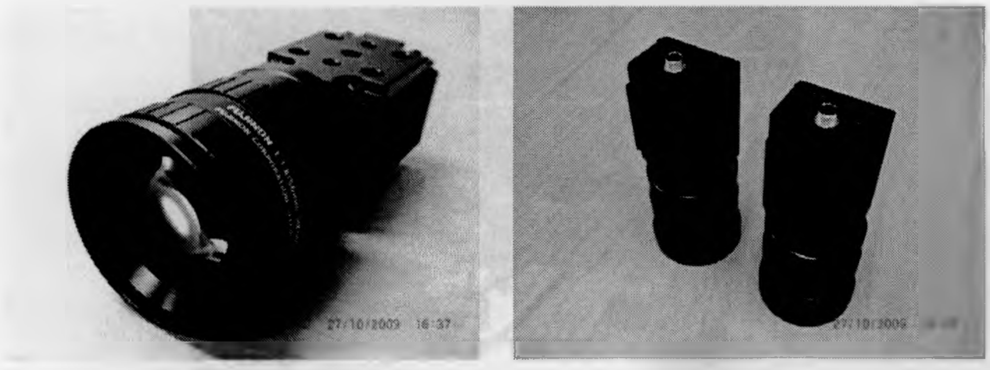
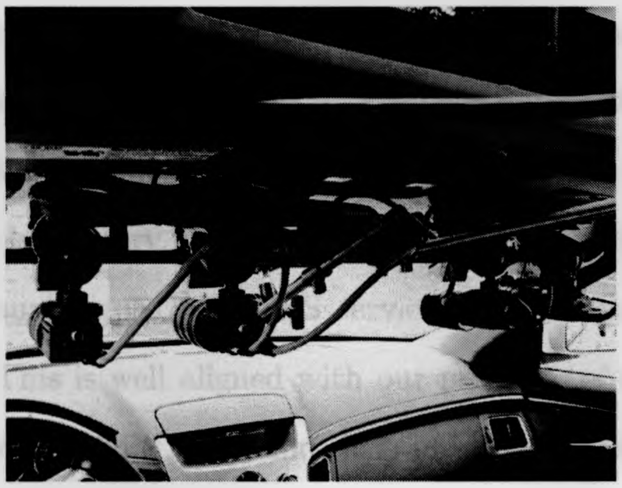


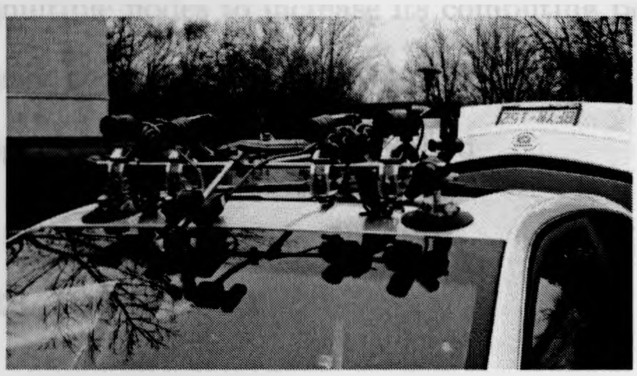
Figure 3.1: *The Grasshopper cameras from PointGrey were used for the RoadLab ADAS instrumentation. The cameras can operate on either on a 1394a or 1394b firewire bus. The cameras are capable of using a number of different lens types. Currently, we are using one pair with 12.5mm and the other pair with 25mm lenses on a 1394a firewire bus.*

ical to obtain precise calibration parameters such as lens distortion, the optical center, and the external orientation of sensors with respect to each other. This calibration is necessary to perform stereo computation and to estimate distances of objects (other vehicles, pedestrians, etc.), which in turn greatly simplifies other vision-related tasks such as estimating motion, tracking, and detecting obstacles.

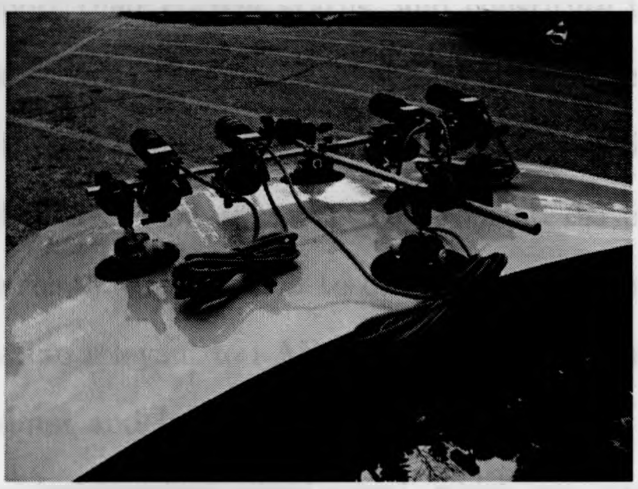
Due to the nature of instrumentation on the vehicle, a relatively large amount of effort is put into minimizing the vibration during the hardware design phase, and professional automotive camera mounts are used. The visual sensors are mounted on a single metal rod with vacuum devices attached at the both ends, and an additional rod with the vacuum device is connected to the rod with sensors to triangulate the mount and minimize the vibration (see Figure 3.2). In this way, cameras can be mounted on any of the vehicle's horizontal surfaces. We have tested mounting the visual sensors on the top of the vehicle, on the hood of the vehicle, and on the windshield inside of the vehicle (see Figure 3.2). The setup gives us great flexibility in the mounting position without additional effort.



(a) Mounting the visual sensors on the windshield inside of the vehicle



(b) Mounting the visual sensors on the top of the vehicle



(c) Mounting the visual sensors on the hood of the vehicle

Figure 3.2: *Different Experimental Configuration with Cameras.*

3.3 GPS unit

The utilization of the Global Positioning System (GPS) is another important aspect of ADAS [11]. ADAS applications require an accurate knowledge of road structures. The

GPS data includes location information including altitude, latitude, and longitude, as well as the course and the velocity of the vehicle. The information from a GPS unit in combination with the map data becomes a powerful tool for ADAS.

For our in-vehicle laboratory, the GlobalSat BU-353 GPS unit is used. The GPS data is obtained through "gpsd"[1], a GPS service daemon that provides an event-driven architecture. This is well aligned with our principle of scalability. gpsd supports numerous vendors of GPS units since it is independent of the driver. Another advantage of gpsd is the capability of socket communication. The data processing unit may consist of multiple nodes to increase its computing power. In a such case, GPS data can be easily accessed by other ADAS components from different nodes without implementing a new communication method.

3.4 CANbus interface

Contemporary vehicles equipped with On-Board Diagnostic systems (OBD-II) allow vehicle sensors to report their current status, and constitute the interface through which odometry is made available in real time. Since 2008, the CANbus protocol is mandatory for OBD-II. This standardization simplifies the real-time capture of vehicle data. OBD-II to USB hardware interfaces with appropriate drivers are now commonly used to route vehicle-related information to on-board computers or similar devices. The available information relevant to i-ADAS applications include current speed and acceleration (longitudinal and lateral), steering wheel rotation, state of accelerator and brake pedals, and independent wheel speed, which are captured in real-time at frequencies between 20 and 200Hz. These elements provide the information that is required to understand the manoeuvres affected by the driver.

The data from the OBD-II/CANbus is accessed by creating a software layer. Additionally, the incoming data from the instrumentation includes timestamps, allowing

[1]Available at http://gpsd.berlios.de/

the system to fuse and select data elements in a synchronized fashion.

In our instrumented vehicle, two types of bus speeds are available. Different types of information are gathered through these buses. High-speed CANbus contains information on the movement of the vehicle: brake pedal position, steering wheel position, individual wheel speed, engine speed, etc. Low-speed CANbus interface collects the state of interior controls: signals, wipers, remote radio, phone controls, hazard lights, fans, seatbelts, etc.

All the information gathered from the CANbus interface is valuable: such information is a great indication of the vehicle state as well as a driver's operation of the vehicle.

3.5 Data processing unit

On-board computing capabilities must be sufficient to process incoming data in real time. To this end, we have designed and assembled a computer for real-time data processing and fusion consisting of 16 cores, each running at 3.0GHz, with 16GB of internal memory and a 128GB Solid State Drive (SSD), with Linux Debian 5.01 as the operating system (see Figure 3.3). The nodes are networked with a high-end gigabit network switch, and configured as a disk-less cluster, with the master node providing the operating system image to other nodes. By our design choice, it is easy to add extra nodes to increase the number of cores and the computing power as necessary. In addition, each node can be equipped with a CUDA-enabled graphics card[2] to take advantage of GPU's parallel architecture.

The data processing unit and instrumentation are powered with a 1500W inverter connected directly to the battery of the vehicle. The instrumentation can run continually without battery drainage. To protect the devices from the noise-contaminated power, we have employed a power conditioner. The power conditioner cleans and

[2]CUDA is NVIDIA's parallel computing architecture

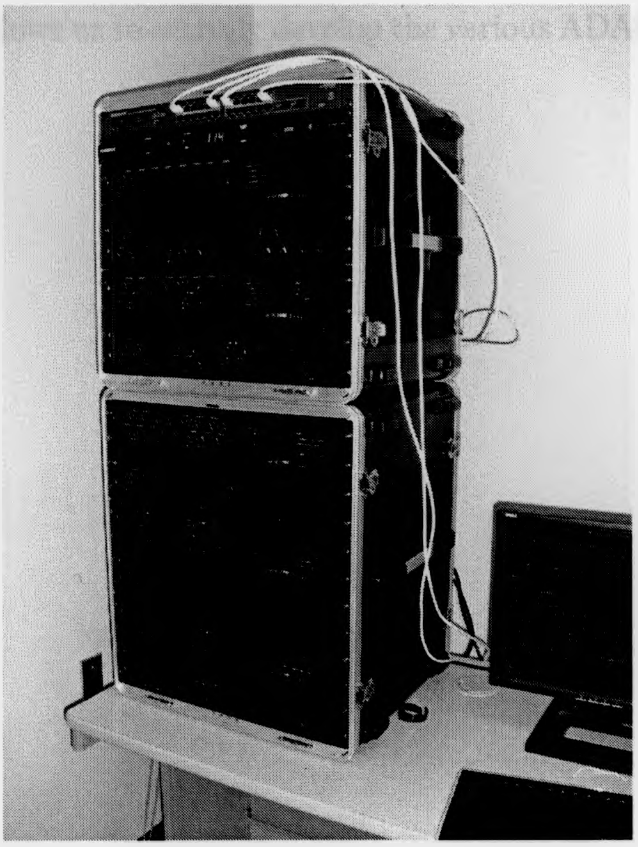


Figure 3.3: *The Data Processing Unit for RoadLab: The processing unit in the image operates with four computing nodes. The top node operates as a master to provide an operating system image to other nodes. Only the master node is equipped with a hard disk. Other nodes are configured to operate as a disk-less cluster.*

filters incoming power and increases the durability of the system.

3.6 The Development of i-ADAS

The in-vehicle laboratory is used as a platform to collect the sequence of the road images and test different ADAS modules. The recorded sequence shall cover various scenarios on the road. Highway, urban area and rural road sequences are required in various weather conditions at different times of the day. Since the change of light sources and other illumination problems are challenges that the vision-based system must overcome, it is critical to obtain such sequences as a part of the dataset.

The dataset provides rich information about the vehicle state and the surrounding environment. This allows us to actively develop the various ADAS related applications and modules.

Chapter 4

Data Calibration

4.1 Camera Calibration

The decline in the cost of computer and camera devices has resulted in a growth in stereo-based applications. For visual sensors, it is critical to obtain precise calibration parameters, such as lens distortion, the optical center, and the external orientation of sensors with respect to each other. Again, calibration is required to perform stereo and to estimate distances of objects (other vehicles, pedestrians, etc.), which in turn greatly simplifies other vision-related tasks such as estimating motion, tracking, and detecting obstacles. The RoadLab stereo calibration interface is created for this process.

4.1.1 Calibration Techniques

There are two main categories of camera calibration: photogrammetric and self-calibration. A photogrammetric method is performed by observing a geometrically well-known calibration object in 3-D. The calibration objects are usually constructed from two or three orthogonal planes or a single plane with a checkerboard pattern embedded to the surface. The most well-known technique from this category is Tsai's method from 1987 [35]; despite the age of this method, the method delivers high precision and therefore is very widely used.

The term self-calibration, or auto-calibration, is used when no calibration object is required. Instead, images taken by moving a camera in a rigid scene and other constraints on the internal camera parameters are used. Self-calibration was first introduced by Maybank et al. [22]. Since Maybank's proposal, the self-calibration method has become an active research topic in the Computer Vision community. The self-calibration method is very flexible. However, due to the large amount of parameter estimation required by this approach, it is difficult to produce a precise outcome. The results of such methods are not yet comparable to the ones obtained from photogrammetric methods.

The hybrid of the two methods above was suggested by Zhang [39], and became very popular in the Computer Vision community. Zhang's method permits a more flexible configuration of the calibration object without degrading the accuracy.

The researchers have conducted the number of reviews to assess the existing calibration methods. The recent literature reviews conducted by Sun and Cooperstock [32] and Salvi et al. [28] provide details of the traditional calibration methods. Salvi *et al.* make the comparison between methods from Tsai [35], Hall [12], and Faugeras and Toscani [10] to obtain the most accurate method. Sun and Cooperstock provide another comparison between methods from Tsai [35], Heikkila [13] and Zhang [39]. All the techniques reviewed by Sun and Cooperstock, and Salvi *et al.* are popular and widely used in the Computer Vision community. The result of the surveys indicated that Tsai's method achieved the highest performance.

The review from Sun and Cooperstock uncovers a drawback of Tsai's method, however. This drawback is that the result of the calibration is highly sensitive to the setup of the calibration. To achieve accuracy, one must precisely configure the calibration environment, which requires a vast amount of time and effort. This is true for any conventional method that is based on a world reference approach. Zhang's method is less prone to noise from the calibrating configuration.

It is also worth noting the iterative nature of above methods. Clearly, there are existing non-iterative models to find the solution, which are faster. The algorithms do not suffer from problem related to convergence and other iterative optimization-related issues. However, the nature of noise on the image significantly reduces the accuracy. Therefore, the iterative refinements of the methods are the main factors in determining accuracy. We can clearly see the tradeoffs between accuracy and flexibility.

As Zhang's title, "A flexible new technique for camera calibration," suggests, this calibration method can be easily performed in casual fashion with acceptable accuracy[1]. This is well suited for the purpose of RoadLab. The position of visual sensors in the in-vehicle laboratory can vary from time to time. The visual sensors can be equipped with different ranges. By adopting Zhang's method, we can minimize the loss of accuracy and maintain great flexibility.

4.1.2 Camera Calibration Process

In a typical camera calibration process, there are two types of parameters that need to be discovered. One is called the intrinsic parameters. This intrinsic parameters describe the internal geometry and optical characteristics of the camera. The extrinsic parameters measure the position and the orientation of the cameras with respect to each other.

Zhang's calibration method requires a planar checkerboard grid images with at least two different orientations. The algorithm uses the extracted corner points of the checkerboard pattern to compute a projective transformation between the image points of the n different images. A homography of each view is then optimized

[1]According to Sun and Cooperstock [32], the acceptable accuracy for Zhang's method is the Normalized calibration error (Proposed by Weng *et al.* [37]) of 2.6.

through the Levenberg-Marquardt algorithm [25], which is a gradient-descent method to minimize functions. The images of the two circular points must lie on an absolute conic, which is a point at infinity. This constraint is used to establish a linear system. Using this linear system, some intrinsic parameters are extracted.

Corner extraction from the checkerboard is the key factor for the stable parameter estimation. Once the corners are detected, an additional iterations are conducted to find the sub-pixel accurate location of the corner. The resolution of acquired image frame affects the accuracy of the detection as well. The resolution is set to 640 by 480 pixels. Once the parameters are estimated, the focal length is scaled for 320 by 240 resolutions by division of 2.

In addition, the visual sensors are calibrated in pairs. A pair with the same focal length can be used as another constraint to yield stable intrinsic parameters estimation. The RoadLab stereo calibration interface was designed for this process. The interface is implemented using a calibration algorithm from the OpenCV 2.1 open source library based on Zhang's technique [39]. The calibration process consists of two steps. Intrinsic parameters are first estimated for each sensor and then, based on these, the extrinsic parameters for sensor pairs with the same focal length are obtained. It is also possible to estimate the extrinsic parameters dynamically [8].

4.2 Camera Calibration of multiple stereo pairs

Once the parameters for each stereo pair with the same focal lengths are estimated, additional calibration is conducted. Since we know the intrinsic parameters of the cameras, the calibration for all possible pairs of the stereo that share a common field of view can be easily performed to estimate the relative position with respect to each stereo pair. The same calibration interface is used as the previous section, which is based on Zhang's method. This allows us to set all the cameras on the same coordinate system.

For the cameras that do not share a common view, additional information is needed. One way of calibrating is to use a special calibration tool so that the stereo pair can view the same object of known size to calculate the relative position of cameras. Another alternative is to detect the most distinct features around the environment and track them (e.g. [21], [40] and [16]). The calibration of such a type of configuration will be included in the future system.

4.3 Data synchronization

The data from visual sensors must be synchronized with data from CANbus interface and GPS unit. Each interface creates a timestamp based on the system time. For recording, the data acquisition rate for visual sensors will serve as the base. The GPS and CANbus interface data stream is collected and attached to the image frame. At an acquisition rate of 30 Hz, all the data gathered after the last image frame up to the current frame is stored together. The data format for the recorded sequence is discussed in detail in chapter 6.

Chapter 5

Stereo Computation

5.1 Stereo Correspondence Algorithm

Disparity maps are processed by matching an object on two or more images with the corresponding elements. Then the range of the object is computed using those detected matches. The method for matching the objects can be categorized as local or global. Local methods search small regions based on the internal characteristics of the given patch. Global methods consider physical constraints such as the continuity of the surface. Disparity maps are widely used in vision-based ADAS modules. The accuracy of the applications greatly depends on the disparity as a result. Due to the hard real-time constraints on these applications, they often rely on local matching algorithms. These algorithms outperform the global optimization method. The local methods can be further classified by feature-based matching or small area patch correlation. Since features (e.g. edges, corners, or blobs) are mostly independent of the lighting and viewpoint, feature-based methods may be more robust than the ones using correlation. However, feature-based matching will result in sparse range disparity[1] with additional time required for computing the feature extraction.

[1]Some parts of images may not have a detectable feature, therefore, resulting a disparity map with missing values

5.2 Stereo Computation in RoadLab

We have deployed a stereo algorithm from OpenCV. The block-matching method in OpenCV for stereo is based on Konolige [17]. Konolige used area correlation to optimize the performance. He uses Laplacian of Gaussian (LoG) transform and L1 norm (absolute difference) correlation. The result of the algorithm is improved further by scaling the image and using multiple resolutions of the same image. In addition, OpenCV uses MMX^{TM} and Strreaming SIMD Extensions (SSE) instruction sets.

Once the disparity map of the scene is estimated, the map can be used in various modules. The distance of the detected object from different ADAS modules is verified using the map; it also provides additional criteria for the detection modules. We have prior knowledge of the size of vehicles, pedestrians, traffic signs or other objects that require detecting. Since we know the distance of the object with respect to the vehicle, we can estimate the size of detected object. The estimated size can be used to distinguish the outliers.

The estimated values of disparity allow us to calculate the ground plane as well. By extracting the ground from the image frame and removing unnecessary information (eg. objects above the ground), it is possible to build a proper region of interest in the image (e.g. search only bottom half of the image, which is the ground, for detecting lane markings). This significantly reduces the computation time and brings us one step closer to real-time programming.

Chapter 6

Data format and acquisition

A RoadLab research film sequence, which is stored in a folder, consists of XML files and RDS files that are used to represent and interpret the data collected by the instrumented car. The number of files will vary with the length of the filming session and the number of cameras used.

The XML files contain camera parameters used to calibrate the cameras for use as stereo pairs. The RDS files contain an amalgamation of all the data collected by the system at the time of recording. The RDS file names are auto-generated by the system and consist of sequential 10 digit numbers (with leading zeros). The RDS file contains a header and a data section. The header contains information related to the images and cameras that are used to interpret the data section. The data section is a series of data frames. Each data frame contains GPS data, CANbus data and the image data. Since the data from the CANbus messages have different frequencies, all the CANbus data from the last image acquisition to the current image acquisition is stored in each data frame. Therefore, an additional data header is required for each data frame to know the size of the CANbus data .

The number of image frames in the data frame can also vary. Since the system is easily extendable, different numbers of cameras can be used. The header of the

RDS file contains the information acquired before reading the data (Size of the image frame, number of the cameras used during recording sessions, colour information, and frame rate)(see Figure 6.1). To avoid data loss, there is no compression in the image frames, and a bitmap format is used. The image frames can be stored in RGB or gray-scale. At the time of writing, we are collecting image frames at 30Hz with 320 by 240 pixels.

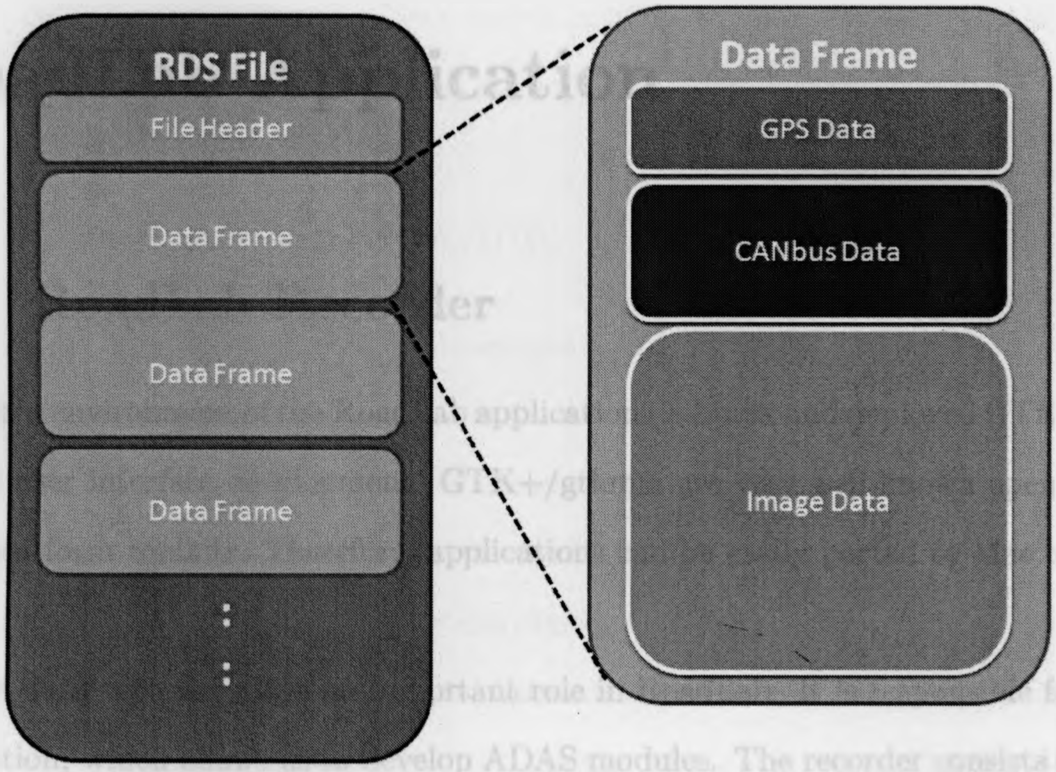


Figure 6.1: *RDS File Structure: The file header consists of the image acquisition frame rate and the image format. The data frame is created for every frame. Each data frame contains GPS data at the time of the image acquisition, the CANbus messages since the previous frame. The images in the data frame are stored in a sequential manner based on the ID number of the camera on the firwire bus.*

The camera parameters are stored in XML format. The intrinsic and extrinsic parameters are stored in separate files for each camera and each camera pair. Therefore, if the configuration of the system is consistent, one can easily replace the parameters from different calibration tools if necessary.

Chapter 7

RoadLab Application

7.1 RoadLab Recorder

The main environment of the RoadLab applications is Linux and deployed GTK+/gtkmm[1] for the user interface development. GTK+/gtkmm are very well known open source cross-platform toolkits. Therefore, applications can be easily ported to Mac or Windows.

The data recorder plays an important role in RoadLab. It is responsible for data acquisition, which allows us to develop ADAS modules. The recorder consists of multiple threads. Each thread is responsible for communicating with hardware interfaces and writing the input data to the hard drive.

We have to minimize the data loss during filming. The bottleneck for recording the sequences is writing data on the physical media. The common procedure for writing data on the hardware is to read data from the interface, and then buffer the data, and finally, write it on the actual storage medium. Increasing the number of write calls can reduce the system performance dramatically.

In order to reduce the number of write calls, we have employed a cyclic queuing mechanism. All the acquired data is first queued onto the memory. Since the speed

[1]Toolkit for creating graphical user interfaces, more detail available at http://www.gtk.org/

of memory is significantly higher than the writing speed on the physical medium, the delay is minimized. We have implemented the queue in a way that en-queuing can be possible while de-queuing is in progress, and vice versa. The writing call is threaded, as we have mentioned, to keep any blocking the data acquisition process to a minimum.

In addition, the recorder provides a way to view stereo computation in order to quickly test the sequence and allows encoding of image data to compress the data if necessary (see Figure 7.1).

7.2 RoadLab Calibration

The calibration software allows us to estimate and store the intrinsic and extrinsic parameters for available cameras. Prior to the calibration process, we need to know the number of columns and rows on the calibration board with a checker pattern, along with its width and height. If the intrinsic parameters of the selected camera were already available, then the user can choose to calibrate the extrinsic parameters only. When the chessboard pattern is presented in the scene, the calibration software will detect the number of corners. If the number of corners were valid, it allows the user to save the image frame. Once the user is collected more than two pairs of images, he/she can run the calibration, that estimates the camera parameters and saves these parameters in XML format. The images used in the calibration can be stored and can be used as well (see Figure 7.2).

Once the calibration is done, users can test the result by rectifying current cameras or by computing disparity. From our experience, we have discovered that for reliable parameter estimation, the calibration image set should cover all areas of the image.

The calibration tool adds great flexibility to the RoadLab instrumentation. The location of visual sensors can be changed as often as needed for the desired coverage

Figure 7.1: *RoadLab Sequence Recorder: The recorder either allows a user to view two pairs of images in real-time, or from a previously recorded session. Users can choose the viewing mode of the pair, selecting from raw, rectified, or stereo. The raw mode displays the images directly from the cameras or the recorded file. The rectified mode shows horizontally aligned images of the selected pair (using the calibration parameters.) The stereo mode computes a disparity map and shows the result. The user can change the view mode at anytime during the recording and playback sessions. The top pair of images above uses raw mode and the bottom pair stereo mode.*

of the surroundings.

Figure 7.2: *RoadLab Calibration Interface: The calibration interface allows us to calibrate the cameras and store the parameters in XML files.*

7.3 RoadLab Reader

The goal of the reader application (see Figure 7.3) is to allow any users to download and test the sequence. Therefore, the reader supports multiple platforms including Windows, different Linux distributions and OS X. The source code of the reader is available with recorded film sequences on the web.

The rectification and depth map computation features are integrated into the reader so that other researchers can use the 3D information without additional effort. The plan for the reader is to include the ability to add functionalities as a module, so that users can easily add their own functions to test their results rather than modifying the reader code.

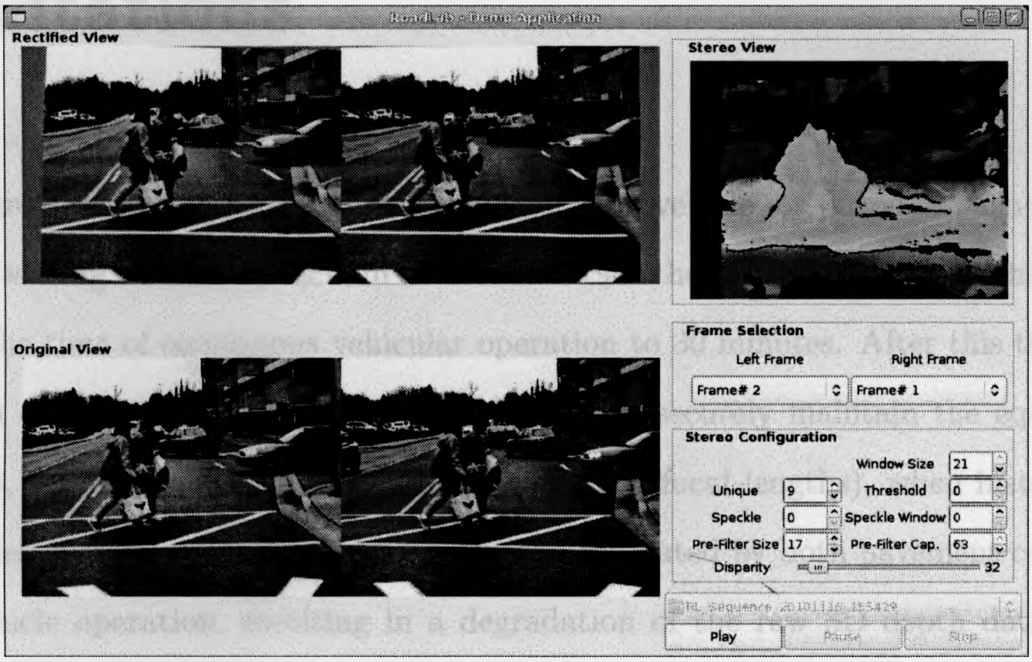


Figure 7.3: *RoadLab Reader: Once the road image sequences are stored using the recorder, data can be viewed using the reader. On the reader, users can select the desired camera pair and view the recorded sequence. The reader shows the images directly from files, as well as the rectified images and a computed disparity map.*

Chapter 8

Limitations

There are several limitations with instrumenting a vehicle for purposes such as ours. On the sensing side, using vacuum devices to attach the instrumentation to the vehicle limits the time of continuous vehicular operation to 30 minutes. After this time, the vacuum device pumps must be operated again to securely maintain the equipment in place. In addition, long-range lenses (with long focal lengths), when installed on the stereo systems, are sensitive to vibrations generated by both pavement condition and vehicle operation, resulting in a degradation of the raw 3D depth data. This problem is worse when the mounting configuration is located inside the windshield, as it introduces distortions that cannot be easily corrected.

The availability of on-board computing power is inherently limited by the available space and electrical power in the vehicle. For instance, the use of high-resolution cameras would severely compromise our requirements for frame-rate processing. In this case, the problem may be addressed by replacing the computing nodes with GPUs, involving significant material costs. There is also the possibility of vehicle battery drainage with the use of high-end computing equipment, requiring the installation of a high-output, after-market vehicle alternator. In addition, our use of solid state drives limits the amount of time the vehicle can be operated in recording mode. In our case, this limit is between 10 and 30 minutes, depending on how many visual sensors are in use while recording.

Chapter 9

Performance Evaluation of Platform

The dual stereo systems constitute an essential component of the instrumented vehicle and, for this reason, their performance (related to raw 3D depth data) is crucially important. We first consider the problem of range resolution, which is inversely related to object distance. The relationship governing range resolution is given by

$$\Delta r = \frac{r^2}{bf}\Delta d \tag{9.1}$$

where r is distance to object; f, focal length of imaging lens; b, stereo baseline length; and Δd, pixel size divided by the interpolation factor of the epipolar scan-line algorithm (for sub-pixel-precision 2D matching). The range resolutions for our dual stereo systems constitute a reliable indication of the error levels contained in the depth data, provided that calibration is accurate and that the depth measurements do not stem from incorrect 2D matches (due to occlusion, spatial aliasing, image noise, or related problems). Many dense stereo vision algorithms have been comparatively evaluated (including that of OpenCV, which we use) with image sequences for which true depth is available in terms of incorrect match density and resilience to noise [33]. The short-range stereo system has a baseline of length $b = 357$ mm, a smallest detectable 2D disparity of $\frac{1}{16}$ of a pixel, a focal length of $f = 12.5$ mm, and a physical pixel square size of 4.40 µm. The long-range stereo system differs only in its baseline ($b = 678$ mm) and focal length ($f = 25.0$ mm). Figure 9.1 shows the range resolution functions for both stereo systems. As expected, the range resolution of the long-range stereo pair surpasses that of the short-range, due to an extended baseline and a longer focal length of the lens.

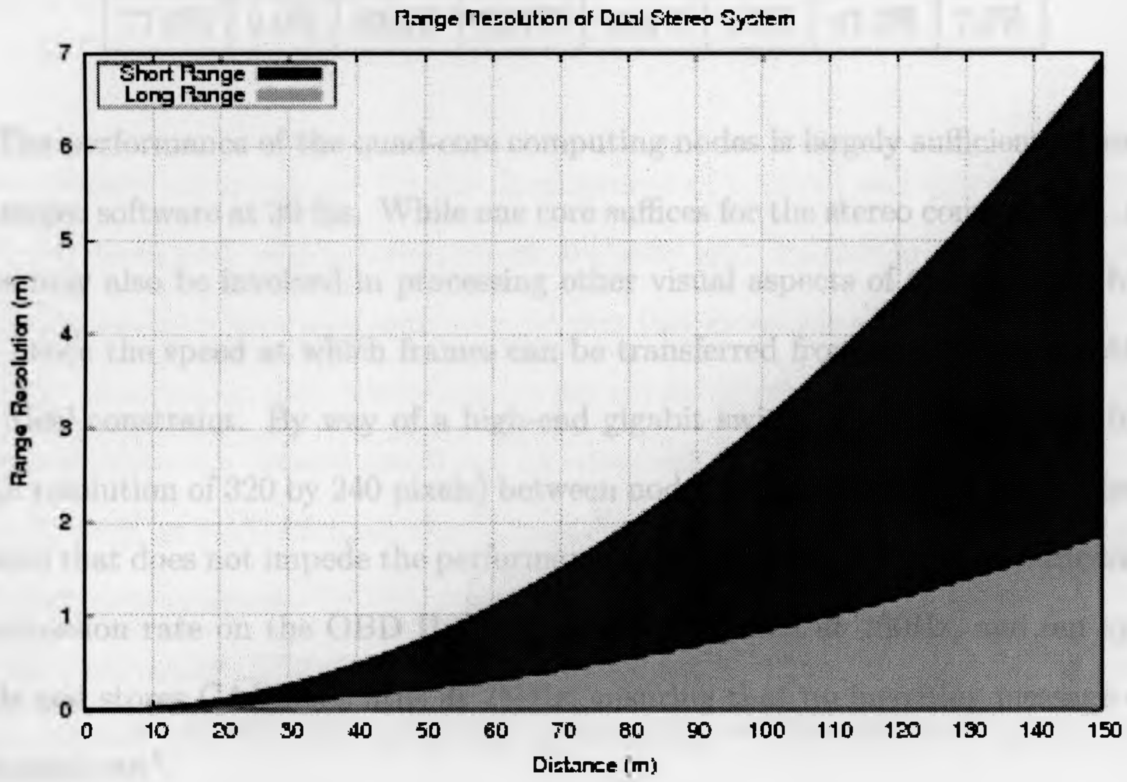


Figure 9.1: *Range resolution functions for dual stereo system, from* 0 *to* 150 m. *As shown, the error rate of the stereo system will exponentially increase with respect to the distance*

We have computed the average match density[1] of both the long- and short- range stereo systems using instrumented sequences produced with the vehicle on public roads[2]. Results are reported in Table 9.1, where different values of the minimum disparity[3] were used. As can be observed, the short-range stereo system performs

[1]The number of matched pixels out of the size of density map is averaged.

[2]The instrumented sequences used to perform these computations are publicly available at www.csd.uwo.ca/faculty/beau/roadlab_download/index.html.

[3]The minimum disparity parameter controls the offset to the disparity search window. Increasing

better in terms of density, due to several factors, including the reported fact that operational vibrations introduce more noise in long-range systems.

Table 9.1: *Stereo match density for short and long-range systems, where d is minimum disparity and D is match density with standard deviation σ.*

Stereo Average Density							
Short-Range				Long-Range			
$d = 32$		$d = 64$		$d = 64$		$d = 96$	
D	σ	D	σ	D	σ	D	σ
71.6%	9.0%	82.5%	10.1%	49.4%	7.7%	41.3%	7.5%

The performance of the quad-core computing nodes is largely sufficient to execute the stereo software at 30 fps. While one core suffices for the stereo computation, other cores may also be involved in processing other visual aspects of the captured frames and hence the speed at which frames can be transferred from one node to another is a critical constraint. By way of a high-end gigabit switch, the cores transfer frames (with resolution of 320 by 240 pixels) between nodes at 1.4MHz (or 0.7 ms per frame), a speed that does not impede the performance of the system. Additionally, the highest transmission rate on the OBD II CANbus was measured at 200Hz, and our system reads and stores CANbus status at 2MHz, ensuring that no incoming message could be missed out[4].

positive values have an effect similar to augmenting the convergence of the stereo cameras.

[4]Performance ratings of other aspects of our instrumentation such as the GPS device (GloablSat BU-353) is published by manufacturers and not reported herein.

Chapter 10

Conclusion

The complete infrastructure for the development of i-ADAS was described in detail. We have developed vehicle-independent, portable and scalable in-vehicle instrumentation for i-ADAS. Our motivation to develop this in-vehicle research platform stems from the observation that while injuries per driven kilometre are in decline in developed countries, the reverse trend can be observed elsewhere in the world [1]. Technologies such as i-ADAS have the potential to significantly reduce vehicle accidents and their consequences.

We have made the datasets collected from our platform available to the public. These dataset can be read through the applications we have provided, which are also available to the public. The reader application can be easily modified to utilize the data as one needs. Researchers can save tremendous amounts of time in initializing their projects, and focus on their new methods. Furthermore, researchers with existing ADAS can test their performances before their actual road test.

In future work, the calibration algorithms for vision sensors that do not share a common field of view will be more closely investigated and integrated into the platform. The visual data of the driver may be added to the dataset as well.

Bibliography

[1] N. H. T. S. Administration. Traffic safety facts 2006: A compilation of motor vehicle crash data from the fatality analysis reporting system and the general estimates system. *Tech. Rep. DOT HS 810 818*, 2006.

[2] A. Amditis, E. Bertolazzi, M. Bimpas, F. Biral, P. Bosetti, M. Da Lio, L. Danielsson, A. Gallione, H. Lind, A. Saroldi, et al. A holistic approach to the integration of safety applications: The INSAFES subproject within the European framework programme 6 integrating project PReVENT. *Intelligent Transportation Systems, IEEE Transactions on*, 11(3):554–566, 2010.

[3] S. Ammoun, F. Nashashibi, and C. Laurgeau. An analysis of the lane changing manoeuvre on roads: the contribution of inter-vehicle cooperation via communication. In *Intelligent Vehicles Symposium, 2007 IEEE*, pages 1095–1100. IEEE, 2007.

[4] S. Ashley. Crashless cars: Making driving safer. *Scientific American Magazine*, 2008.

[5] K. Ball, C. Owsley, B. Stalvey, D.L. Roenker, M.E. Sloane, and M. Graves. Driving avoidance and functional impairment in older drivers* 1. *Accident Analysis & Prevention*, 30(3):313–322, 1998.

[6] M. Bertozzi and A. Broggi. GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection. *Image Processing, IEEE Transactions on*, 7(1):62–81, 2002.

[7] A. Bose and P. Ioannou. Analysis of Traffic Flow With Mixed Manual and Intelligent Cruise Control Vehicles: Theory and Experiments. *Institute of Transportation Studies, Research Reports, Working Papers, Proceedings*, 2001.

[8] J.M. Collado, C. Hilario, A. de la Escalera, and J.M. Armingol. Self-calibration of an on-board stereo-vision system for driver assistance systems. In *IEEE Intelligent Vehicle Symposium, Tokyo, Japan*, pages 156–162, 2006.

[9] ED Dickmanns, R. Behringer, D. Dickmanns, T. Hildebrandt, M. Maurer, F. Thomanek, and J. Schiehlen. The Seeing Passenger Car VaMoRs-P, Intelligent Vehicles 94 Symposium. In *Proceedings of the*, 1994.

[10] O.D. Faugeras and G. Toscani. The calibration problem for stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 86, pages 15–20, 1986.

[11] S. Glaser, S. Mammar, and C. Sentouh. Integrated Driver–Vehicle–Infrastructure Road Departure Warning Unit. *Vehicular Technology, IEEE Transactions on*, 59(6):2757–2771, 2010.

[12] E.L. Hall, J.B.K. Tio, C.A. McPherson, and F.A. Sadjadi. Measuring curved surfaces for robot vision. *Computer*, 15(12):42–54, 1982.

[13] J. Heikkila. Geometric camera calibration using circular control points. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(10):1066–1077, 2000.

[14] U. Hofmann, A. Rieder, and E.D. Dickmanns. Radar and vision data fusion for hybrid adaptive cruise control on highways. *Machine Vision and Applications*, 14(1):42–49, 2003.

[15] A.S. Huang, D. Moore, M. Antone, E. Olson, and S. Teller. Finding multiple lanes in urban road networks with vision and lidar. *Autonomous Robots*, 26(2):103–122, 2009.

[16] O. Javed, Z. Rasheed, K. Shafique, and M. Shah. Tracking across multiple cameras with disjoint views. 2003.

[17] K. Konolige. Small vision systems: Hardware and implementation. In *Robotics Research-International Symposium*, volume 8, pages 203–212. MIT PRESS, 1998.

[18] Y. Liang, M.L. Reyes, and J.D. Lee. Real-time detection of driver cognitive distraction using support vector machines. *Intelligent Transportation Systems, IEEE Transactions on*, 8(2):340–350, 2007.

[19] J.F. Liu, Y.F. Su, M.K. Ko, and P.N. Yu. Development of a Vision-Based Driver Assistance System with Lane Departure Warning and Forward Collision Warning Functions. In *Computing: Techniques and Applications, 2008. DICTA '08. Digital Image*, pages 480–485. IEEE, 2008.

[20] B. Ma, S. Lakahmanan, and A. Hero. Road and lane edge detection with multisensor fusion methods. In *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, volume 2, pages 686–690. IEEE, 2002.

[21] D. Marinakis and G. Dudek. Topology inference for a vision-based sensor network. 2005.

[22] S.J. Maybank and O.D. Faugeras. A theory of self-calibration of a moving camera. *International Journal of Computer Vision*, 8(2):123–151, 1992.

[23] J.C. McCall and M.M. Trivedi. Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation. *IEEE Transactions on Intelligent Transportation Systems*, 7(1):20–37, 2006.

[24] J.C. McCall and M.M. Trivedi. Driver behavior and situation aware brake assistance for intelligent vehicles. *Proceedings of the IEEE*, 95(2):374–387, 2007.

[25] J. More. The levenberg-marquardt algorithm: implementation and theory. *Numerical analysis*, pages 105–116, 1978.

[26] A. Perez, M.I. Garcia, M. Nieto, J.L. Pedraza, S. Rodriguez, and J. Zamorano. Argos: an advanced in-vehicle data recorder on a massively sensorized vehicle for car driver behavior experimentation. *Intelligent Transportation Systems, IEEE Transactions on*, 11(2):463–473, 2010.

[27] L. Petersson, L. Fletcher, and A. Zelinsky. A framework for driver-in-the-loop driver assistance systems. In *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, pages 771–776. IEEE, 2005.

[28] J. Salvi, X. Armangué, and J. Batlle. A comparative review of camera calibrating methods with accuracy evaluation* 1. *Pattern recognition*, 35(7):1617–1635, 2002.

[29] S. Sekizawa, S. Inagaki, T. Suzuki, S. Hayakawa, N. Tsuchida, T. Tsuda, and H. Fujinami. Modeling and recognition of driving behavior based on stochastic switched ARX model. *Intelligent Transportation Systems, IEEE Transactions on*, 8(4):593–606, 2007.

[30] N. Shibata, T. Terauchi, T. Kitani, K. Yasumoto, M. Ito, and T. Higashino. A method for sharing traffic jam information using inter-vehicle communication. In *Mobile and Ubiquitous Systems-Workshops, 2006. 3rd Annual International Conference on*, pages 1–7. IEEE, 2007.

[31] H. Sugano, R. Miyamoto, and Y. Nakamura. Optimized parallel implementation of pedestrian tracking using HOG features on GPU. In *Ph. D. Research in Microelectronics and Electronics (PRIME), 2010 Conference on*, pages 1–4. IEEE, 2010.

[32] W. Sun and J.R. Cooperstock. An empirical evaluation of factors influencing camera calibration accuracy using three publicly available techniques. *Machine Vision and Applications*, 17(1):51–67, 2006.

[33] H. Sunyoto, W. Van der Mark, and D.M. Gavrila. A comparative study of fast dense stereo vision algorithms. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 319–324. Ieee, 2004.

[34] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, et al. Stanley: The robot that won the DARPA Grand Challenge. *The 2005 DARPA Grand Challenge*, pages 1–43, 2007.

[35] R. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, 1987.

[36] C. Urmson, J. Anhalt, M. Clark, T. Galatali, J.P. Gonzalez, J. Gowdy, A. Gutierrez, S. Harbaugh, M. Johnson-Roberson, H. Kato, et al. High speed navigation of unrehearsed terrain: Red team technology for grand challenge 2004. *Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-04-37*, 2004.

[37] J. Weng, P. Cohen, and M. Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on pattern analysis and machine intelligence*, pages 965–980, 1992.

[38] G.R. Widmann, M.K. Daniels, L. Hamilton, L. Humm, B. Riley, J.K. Schiffmann, D.E. Schnelker, and W.H. Wishon. Comparison of Lidar-Based and Radar-Based Adaptive Cruise Control Systems. *Adaptive cruise control*, page 153, 2006.

[39] Z. Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, 2000.

[40] X. Zou, B. Bhanu, B. Song, and A.K. Roy-Chowdhury. Determining topology in a distributed camera network. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, volume 5, pages V–133. IEEE, 2007.